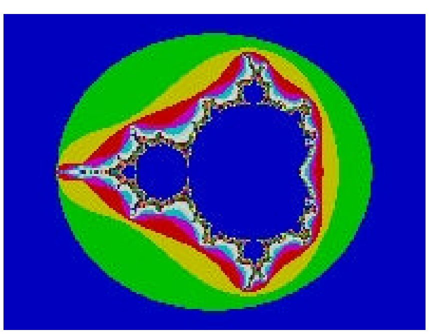
Cloud Visualization Techniques (Some mechanics behind the images)

LT Keith Barto

Since the advent of the first numerical integrators in the 1950’s and subsequent generations of super and micro computers, scientists and computer engineers have struggled with the task of converting computer generated results into formats which accurately convey information and present it in a familiar and understandable form.

The "visual" is the most important medium to clearly communicate information with minimal chance of misinterpretation. Of of our five senses, we comprehend over 80% of the environment around us through the eyes. Use of the 3D representations cross over language barriers, it provides a completely unique experience. (nVIDIA, “The Power of 3D”, 1999)



In the arena of science the various specialties have differing requirements. The mathematician desires to show a parabolic function or render a fractal shape. The cardiologist wishes to probe the heart without cutting open a chest. The meteorologist attempts to show what the weather has been or will become. In meteorology however, anyone who ever stepped outside his door has seen the sky on a sunny day or during a rain shower and understands what things are supposed to look like during differing types of weather. Representing weather systems in formats which end users are familiar has become a new science in and of itself.

When once the next generation of computer was anxiously awaited so a model could run with more data or parameters, now in tandem, graphic workstations and image processing systems continue to evolve allowing automated output to be rendered in ever more lifelike detail.

The most recognizable elements of weather are clouds. Rendering clouds presents numerous challenges. Some of the factors which must be considered are:

- The composition of the cloud forms, their thickness, and shape.
- The 3 dimensional parameters of the atmosphere.
- The perspective of the viewer
- The orientation of the light source.
- The scale of the display from the microphysical (micrometers) too the planetary (thousands of kilometers).
- Motion with respect to time.

For most of the eighties and into the nineties, virtual reality and real-time graphics applications have been implemented on platforms specifically developed for the task. Silicon Graphic has been the primary computer platform for real-time graphics systems because they have designed hardware specific to the needs to 3D rendering and placed that hardware in the boxes they build.
The price of the technology required to do this has been coming down over the last few years,
and in many situations the ability to perform high speed computer graphics has moved to the personal computer.

Currently, most virtual worlds are small with medium to low detail models. Part of this is due to the constraints on the systems. Only a specific number of polygons can be rendered per second and exceeding this number with too many high detail models will slow the computer down. However, the general populous sees images of high detail models moving quickly through large worlds in non-real-time productions that are portrayed as if they were real time.

Currently, computers can not store and display high detail worlds the size of the Earth. Because of this, certain concessions must be made. In computer science, the choice that must often be made is between space and speed. The more memory a process takes, the faster it can go. The challenge for computer scientists is finding ways to use less space without sacrificing too much speed. In general, speed is not something that can be sacrificed. Because these are real-time applications, the algorithms must be fast enough to cause no visible slowing in the rendering.

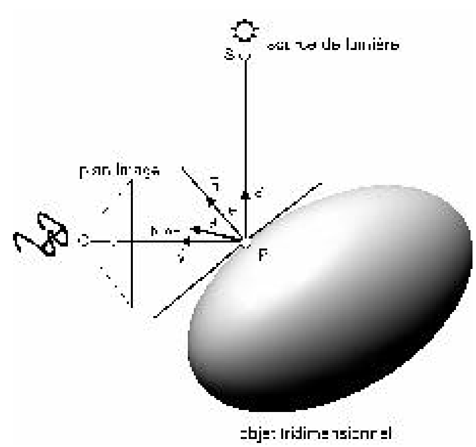
Two different techniques have been developed over the past years for the visualization of volumetric data, allowing for extracting different kinds of information from the 3-D data sets.

The first technique is known as contouring [Lorensen, 1987], where all values in the data set that are below a specified threshold are discriminated. This results in a discrete, iso-valued surface called contour. This surface can be rendered into a 3 dimensional perspective by using shading techniques.

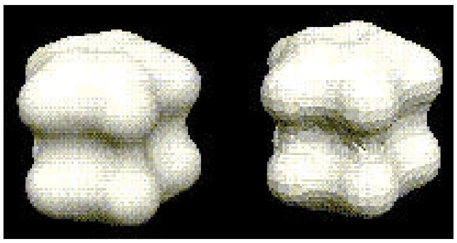
The intensity (color or gray-scale) at each pixel, corresponding to a polygon, is calculated by attributing one color per facet, by scalar interpolation (Gouraud) or vectorial interpolation (Phong).

$$I = k_a \cdot I_a + \frac{k_d \cdot \cos\theta + k_s \cdot \cos^n \beta}{d_s + k_{att,1}} \cdot I_s$$

For the following example:

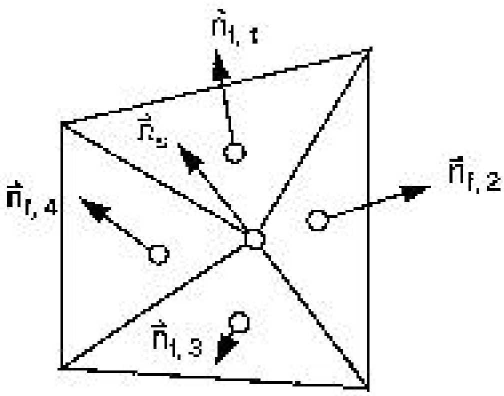


Using the constant shading technique, each facet of the object is illuminated by an average value for the entire polygon. This approach is fast and very simple, but it gives quite poor realistic results and non smooth surfaces. This is enhanced by the Mach effect: the intensity at the vicinities of the edges is overestimated for light values and underestimated for dark values.

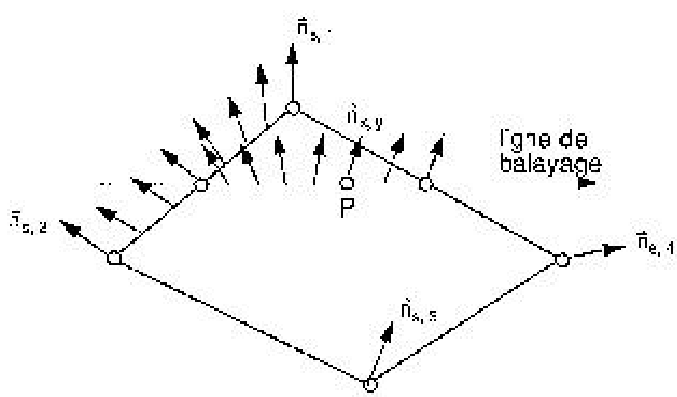


The picture on the left shows a Connolly surface of ferrocene in Gouraud shading, and on the right the same surface with flat shading.

The Gouraud shading technique eliminates intensity discontinuities by interpolating the intensity for each polygon. It uses the normal vector at each vertex and edges of the polygon mesh (obtained by averaging each normal of the facets sharing the same edge). The model determines the intensity at each vertex and then interpolates linearly between each normal along the edge and then the same way between the edges for every scan-line. This scan-line algorithm is very often hardware implemented. The Mach effect is almost completely eliminated (except for very high curvated surfaces).



Phong shading is similar to Gouraud shading based on an interpolation algorithm except that this time, the interpolation is made by vectors. It uses the normals at each facet, the average normals at each vertex, and interpolates vectorially along the edges between the vertex, and then interpolates the same way between the



edges along the scan-line (very heavy calculation, as it has a normalization calculation at every step). This model gives a nice render to specular lights.

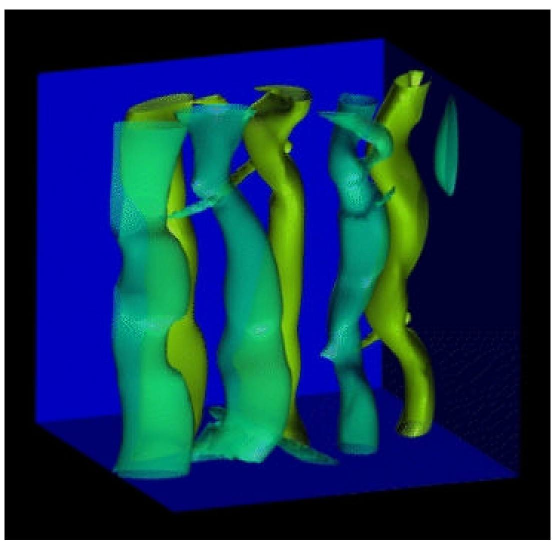
The most common problems encountered with interpolated shading are overcome by utilizing triangles as polygon or by enhancing the numbers of polygon (which is expensive).

For example, a highly curved surface (typically a sphere) will clearly have a polygonal silhouette. This situation can be improved by breaking the surface into a greater number of smaller polygons, but will be more expensive.

The second technique, known as volume rendering [Drebin, 1988], treats the entire data set as a contiguous density cloud and is visualized by modulating the opacity of the object based on the values present in the data set.

Glyphs (geometric obects)can be scaled and colored according to the magnitude of the related data value. By interactive modification of an opacity map, ranges of data values may be made transparent or semi-transparent, allowing the identification of coherent structures within the data. Vector glyphs are oriented in 3-D space to depict the direction of the flow through the volume of a vector quantity. Stereo viewing and animation provide a full 3-D representation of the structures within the data volume.

Using clouds of normally distributed random points of varying numerical density and variance representation of scalar fields is made possible. This technique allows very rapid display. Many rotations and other interactive manipulations are possible without noticeable delay. The technique also allows the quick display of turbulent or noisy quantities that would have highly involuted isosurfaces, and provides much of the information conveyed by nested isosurfaces without the need for many partially transparent surfaces and time-consuming calculations.



References:

1. Karron, D.B., Novel Morse Theoretic Tools for Virtual Reality Exploration of Volumetric Datasets: Graphing the Visible Human Project Dataset.
2. Lorensen, W.E., and Cline, H.E. (1987). Marching cubes: a high resolution surface construction algorithm. Computer Graphics 21, 163-169.
3. Drebin, R.A., Carpenter, L., and Hanrahan, P. (1988). Volume rendering. Computer Graphics 22, 65-74.
4. Savary, F., The Representation of Molecular Models Rendering Techniques
5. Roberts, D. A. (NASA/GSFC), Starr, C. (Hughes STX), http://farside.gsfc.nasa.gov/SAVB/SVS/Stereo/